Comparison of Open Source Visual Analytics Toolkits

John R. Harger[a,b] and Patricia J. Crossno[a]

[a]Sandia National Laboratories, PO Box 5800 Albuquerque, NM 87185-1323, USA;
[b]University of New Mexico, Albuquerque, NM 87131, USA

ABSTRACT

We present the results of the first stage of a two-stage evaluation of open source visual analytics packages. This stage is a broad feature comparison over a range of open source toolkits. Although we had originally intended to restrict ourselves to comparing visual analytics toolkits, we quickly found that very few were available. So we expanded our study to include information visualization, graph analysis, and statistical packages. We examine three aspects of each toolkit: visualization functions, analysis capabilities, and development environments. With respect to development environments, we look at platforms, language bindings, multi-threading/parallelism, user interface frameworks, ease of installation, documentation, and whether the package is still being actively developed.

Keywords: Visual Analytics, open source, toolkits, comparison, evaluation

1. INTRODUCTION

Visual Analytics is defined as "the science of analytical reasoning facilitated by visual interactive interfaces",[1] and more specifically "combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets".[2] We define a Visual Analytics Toolkit as both automated analysis functionality and information visualization techniques in an integrated programming library for stand-alone applications or plugin development.

Developers of computer applications employing Visual Analytics have the choice of leveraging third-party toolkits or implementing desired functionality from scratch. There are many advantages in working with toolkits, yet finding a toolkit that combines the desired characteristics for analysis, visualization, and development platform may not be possible. Open source toolkits provide a solution in which packages that come close to developer needs can be extended or modified. Although individual toolkits have been reviewed at websites such as infosthetics.com, no single reference provides developers with the comparative information they need.

2. PRIOR WORK

Evaluations in information visualization have typically focused on comparisons of design elements, usability studies, comparisons of individual visualization techniques and long term case studies.[3,4] There are a number of issues that arise in these types of evaluations, such as determining what parts of a complex system are contributing to the results, bias and subject familiarity in test users.[5] Our evaluation focuses on a high level comparison of visualization and analysis features, in order to give a big picture overview of available open source toolkits.

Laramee[6] provided a practical comparison aimed at assisting developers select the best graphics or visualization library for building their applications through the evaluation of feature sets, software source and development environment, and the ease of implementing a benchmark application. However, the evaluation was limited to an in-depth examination of four pre-selected packages: VTK, Open Inventor, Coin3D, and Hoops 3D. Although we took inspiration from Laramee's approach, we wanted to provide greater breadth by first doing a feature evaluation stage that looked at a wide-ranging set of open source visualization and analysis libraries as a basis for selecting the toolkits used in the implementation stage. Another significant difference is that our evaluation focuses on open source visual analytics toolkits, so we are looking for packages with a combination of visualization and analysis functionality.

Further author information: (Send correspondence to J.R.H.)
J.R.H.: E-mail: jrharge@sandia.gov, Telephone: 1 505 844 7995
P.J.C.: E-mail: pjcross@sandia.gov, Telephone: 1 505 845 7506

3. FUNCTIONAL COMPARISON

This stage compares features of twenty one open source toolkits and applications to narrow them down to a set for deeper analysis. We examined the following toolkits: Axiis,[7] birdeye,[8] Flare,[9] Gephi,[10] The Google Visualization API,[11] GraphViz,[12] Improvise,[13] The Infovis Toolkit (IVTK),[14] The Javascript Infovis Toolkit (JIT),[15] JFreeChart,[16] JGraph,[17] JUNG,[18] NetworkX,[19] Prefuse,[20] Protovis,[21] R,[22] Titan,[23] Tulip,[24] VisAD,[25] WilmaScope,[26] and Zest.[27] The sources for finding these toolkits were the Infovis Wiki (`http://www.infovis-wiki.net/`) and Internet searches.

Sandia National Laboratories, in collaboration with Kitware, Inc., has developed the Titan Informatics Toolkit. Titan is an open source project built upon the scalable architecture of the Visualization Toolkit (VTK).[28,29] Part of the motivation for our evaluation stems from the need to understand how our toolkit compares to other similar packages.

This section describes those areas of functionality that we compared across the toolkits we examined for the first stage of the evaluation. These include visualization functionality, analysis functionality and a description of the development environment(s) used by the toolkit.

3.1 Visualization Functions

Below we list several visualizations that are intended to provide a broad sampling of techniques available. We have broken each category into a set of "views" that support a particular type of visualization. When available, we provide a reference to the paper describing the algorithm(s) used as specified in the toolkit literature. Cells containing only a checkmark (✓) represent either that no specific algorithm was mentioned, which includes simple or trivial algorithms.

Shneiderman[30] identifies seven basic data types in information visualization: 1-, 2-, 3-dimensional, multi-dimensional, temporal, graphs and trees. Tory and Möller[31] describe a different approach to classifying visualizations based on algorithm data types, such as continuous versus discrete values, and further breaking those categories down into the number of dimensions and algorithms on graphs and trees. We have chosen to split our list of visualization algorithms into four categories: graphs, trees, n-dimensional data (both continuous and discrete) and geospatial/spatio-temporal visualizations.

3.1.1 Graph Views

Graphs are any data that contains information that can be represented as nodes or vertices, and edges connecting these nodes. They come in many different forms, so we included several graph view types in order to emphasize which toolkits have a broader visualization capability. We compared *circular*, *radial*, *force directed*, *hierarchical* and *adjacenty matrix* views. Table 1 shows the results of our comparison.

3.1.2 Tree Layout Views

Trees are a special class of graphs that have explicit hierarchical structure which allows them to be laid out in more ways than graphs in general. We chose to compare horizontal/vertical *tree* views, *radial*, *balloon*, *tree map* and *icicle* tree views as shown in Table 2.

3.1.3 Tabular Data Views

The third class of data is a very general one that includes all information that can be represented as the rows and columns of a table. We included common visualizations: *bar charts*, *line charts*, *pie charts*, *stacked charts*, *box plots* and views supporting multidimensional data (such as *scatterplots* and *parallel coordinates*). Table 3 shows the comparison results.

3.1.4 Geospatial and Spatio-Temporal Visualizations

Table 4 compares views that are specialized for viewing geographic information. *Map overlays* are symbols or polygons drawn on top of a map, either a two-dimensional projection or a three-dimensional globe. *Graduated symbol maps* are similar, but the symbols drawn vary in size based on some data set (for example relative populations). *Choropleth maps* use color coding to represent aspects of regions, such as population, crime rate or average incomes.

There are also many different map projections available in modern toolkits. *Dymaxion maps* are global maps projected onto a polyhedron and unfolded into a two-dimensional image. *Cartograms* are two-dimensional projections where the area of a region is controlled by a data value. A *three-dimensional globe* is a textured sphere that may also show terrain data as well as providing additional 3-D overlays (such as graduated symbols and geographic network layouts).

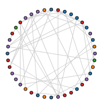
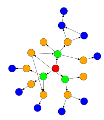
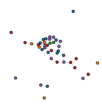
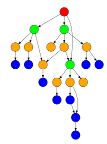
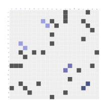
3.2 Analysis Capabilities

Based on our categories in section 3.1, we have divided analysis capabilities into two main categories: analysis on graphs (including trees and networks) and statistical analysis on arbitrary data. Again, we have chosen to focus on general functionality as opposed to individual algorithms, to give a broad overview and comparison of the toolkits.

3.2.1 Graph Analysis

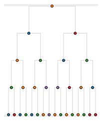
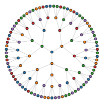
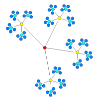
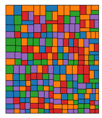
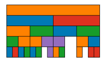
Lee, et al.[47] describe several analysis task categories that can be performed on a graph. These include the following: *Adjacency* tasks involving neighbor vertices, such as finding all neighbors and counting them; *Accessibility*

Table 1. Comparison of Visualization Functionality (Graphs)

	Circular	Radial	Force Directed	Hierarchical	Adjacency Matrix
					
Axiis					
birdeye	✓	✓[32]	✓	✓[33]	
Flare	✓	✓	✓[20,34]		
Gephi	✓	✓	✓[34–36]		
Google Vis					
GraphViz	✓[37]	✓[38]	✓[34,39]	✓[40]	
Improvise[a]	✓	✓[32]	✓[20,34]		
IVTK	✓		✓[34,41]		✓
JIT		✓	✓		
JFreeChart					
JGraph	✓		✓[42]	✓[40]	
JUNG	✓		✓[34,39]		
NetworkX[b]	✓		✓[34]		
Prefuse	✓	✓[32]	✓[20,34]		
Protovis			✓[20,34]	✓[33]	✓
R					
Titan	✓		✓[34]		
Tulip	✓		✓[43]	✓[33]	
VisAD					
WilmaScope[b]			✓		
Zest[b]		✓	✓		

[a] Uses Prefuse for visualization
[b] Also supports using external tool(s) (such as GraphViz) for layouts

Table 2. Comparison of Visualization Functionality (Trees)

	Tree 	Radial 	Balloon 	Tree Map 	Icicle 
Axiis					
birdeye	✓[33]	✓[32]		✓	
Flare	✓[33]	✓[32]		✓[44]	✓
Gephi		✓			
Google Vis				✓	
GraphViz	✓[40]	✓[38]			
Improvise[a]	✓[33]	✓[32]	✓[45]	✓[44]	
IVTK	✓[46]	✓		✓	✓
JIT	✓	✓		✓	✓
JFreeChart					
JGraph	✓[40]				
JUNG	✓	✓	✓		
NetworkX[b]					
Prefuse	✓[33]	✓[32]	✓[45]	✓[44]	
Protovis	✓[33]	✓[33]		✓[44]	✓
R					
Titan	✓	✓		✓[44]	
Tulip	✓[33, 46]	✓	✓[45]	✓[44]	
VisAD					
WilmaScope[b]					
Zest[b]	✓	✓			

[a] Uses Prefuse for visualization
[b] Also supports using external tools (such as GraphViz) for layouts

tasks involving vertices that can be reached from a given node, such as finding all reachable nodes and finding the set of nodes you can reach in a certian number of steps; finally *Connectivity* is a broad category including finding shortest paths, clustering and identifying connected components.

Centrality means calculating measures of relative importance of vertices in a graph. Finally, finding a *minimum spanning tree* for a graph is identifying a tree that is a subgraph containing all of the vertices of the given graph that has the minimum cost in terms of its edges.

3.2.2 Statistical Analysis

Univariate analysis refers to statistical analysis of single-dimensional data, such as finding the mean and variance and calculating histograms. *Bivariate analysis* means statistical analysis of two-dimensional data, such as correlation. *Multivariate analysis* is any sort of analysis of data in three or more dimensions.

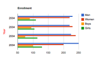
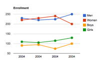
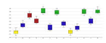
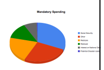
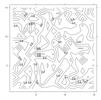
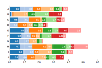
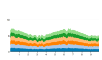
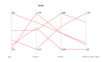
Special purpose techniques include the following: *Dimensionality reduction* and *clustering*. *Dimensionality reduction* is reducing the number of dimensions in multivariate data by removing less significant dimensions. *Clustering* is grouping data points or graph vertices in clusters based on some relatedness criteria.

3.3 Development Environment

A toolkit can be designed to run on Windows, Mac OS X, Unix-like systems, or a virtual machine like the Java Virtual Machine (JVM). The World Wide Web is becoming more popular for deploying visualization applications, and accomplish this using technologies like JavaScript (JS) and Adobe Flex. The *target* column lists the operating system(s) or presentation environment(s) that can be used by the toolkit.

Some toolkits only support applications written in their native *language*, while others also provide interfaces in other languages (possibly high-level scripting languages). Scripting can be a powerful method for prototyping

Table 3. Comparison of Visualization Functionality (Tabular Data)

	Bar Chart 	Line Chart 	Scatterplot 	Box Plot 	Pie Chart 	Contour Plot 	Stacked Bar Chart 	Stacked Area Chart 	Parallel Coordinates 
Axiis	✓	✓	✓		✓		✓	✓	
birdeye	✓	✓	✓		✓		✓	✓	
Flare	✓	✓	✓					✓	
Gephi									
Google Vis	✓	✓	✓	✓	✓		✓	✓	✓
GraphViz									
Improvise[a]	✓	✓	✓					✓	
IVTK	✓	✓	✓						✓
JIT							✓	✓	
JFreeChart	✓	✓	✓	✓	✓		✓	✓	
JGraph									
JUNG									
NetworkX									
Prefuse	✓	✓	✓					✓	
Protovis	✓	✓	✓	✓	✓		✓	✓	✓
R	✓	✓	✓	✓	✓	✓	✓		
Titan	✓	✓	✓		✓				✓
Tulip			✓						✓
VisAD		✓	✓			✓			
WilmaScope									
Zest									

[a] Uses Prefuse for visualization

applications or writing simple one-off programs and these languages can be easier for non-professional programmers to work with.[48] Therefore scripting languages that are directly supported by the toolkits, such as Python and TCL, are also listed here.

GUI frameworks are the front-end interfaces available for each toolkit. These include frameworks such as Qt in C++, Swing and SWT in Java, and web GUI interfaces such as HTML and Adobe Flash.

The *SQL database* column lists any direct support by the toolkit to load datasets from SQL database servers. Some support generic connector APIs (ODBC and JDBC) which can be used to connect to any database system that provides drivers for the connector. Others only support specific databases (MySQL, PostgreSQL, SQL Server) directly through their connection libraries.

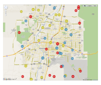
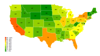
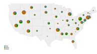
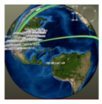
File format support can be useful for exchanging data with other users or applications, especially when dealing with graphs. There are several general exchange file formats, such as comma separated values (CSV), XML and JSON, as well as formats that are designed for graphs, such as DOT, GraphML,[49] GML[50] and VTK's graph file format.[29]

The *documentation* field lists the available documentation, such as tutorials, user's guides, reference manuals and online API documentation.

4. CONCLUSION

Every toolkit has its strengths and its shortcomings. Some toolkits, such as *Protovis* and *birdeye* clearly offer powerful visualization features. Others, like *JGraph* and *JUNG* offer strong analysis functionality. Many of these toolkits focus on one particular area, such as graphs/networks, statistics or geospatial analysis. Some are targeted for a particular platform, such as a web browser. The comparison tables we have provided in this paper can assist users in finding the toolkit that may best suit their needs.

Table 4. Comparison of Visualization Functionality (Geospatial/Spatio-temporal Data)

	Map Overlays 	Choropleth Maps 	Graduated Symbol Maps 	Map Projections 	Dymaxion Maps 	Cartograms 	3D Globe 
Axiis							
birdeye	✓	✓	✓	✓			
Flare							
Gephi							
Google Vis	✓	✓	✓				✓
GraphViz							
Improvise							
IVTK							
JIT							
JFreeChart							
JGraph							
JUNG							
NetworkX							
Prefuse							
Protovis	✓	✓	✓	✓	✓	✓	
R							
Titan							✓
Tulip							
VisAD							
WilmaScope							
Zest							

5. ACKNOWLEDGEMENTS

This work was performed at Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Table 5. Comparison of Analysis Functionality (Graphs)

	Adjacency	Accessibility	Centrality	Connectivity	Minimum Spanning Tree
Axiis					
birdeye	✓	✓			
Flare	✓	✓	✓	✓	✓
Gephi	✓		✓		
Google Vis					
GraphViz	✓	✓		✓	
Improvise					
IVTK	✓	✓		✓	
JIT					
JFreeChart					
JGraph	✓	✓		✓	✓
JUNG	✓	✓	✓	✓	✓
NetworkX	✓		✓	✓	✓
Prefuse	✓				
Protovis					
R					
Titan	✓	✓	✓	✓	✓
Tulip	✓	✓	✓	✓	✓
VisAD					
WilmaScope	✓				
Zest	✓				

Table 6. Comparison of Analysis Functionality (Statistical)

	Univariate	Bivariate	Multivariate	Dimensionality Reduction	Clustering
Axiis					
birdeye					
Flare					✓
Gephi					✓
Google Vis					
GraphViz					
Improvise					
IVTK					✓
JIT					
JFreeChart					
JGraph					
JUNG					✓
NetworkX					✓
Prefuse					
Protovis					
R	✓	✓	✓	✓	✓
Titan	✓	✓	✓	✓	✓
Tulip					✓
VisAD					
WilmaScope					
Zest					

Table 7. Development Environment Comparison

	Target	Languages	GUI	SQL	File Formats	Documentation
Axiis	Web (Flex)	ActionScript	Flash	-	-	API Reference, examples
birdeye	Web (Flex)	ActionScript	Flash	-	GeoRSS, GeoML, KML, SHP	Wiki, API Reference, several examples
Flare	Web (Flex)	ActionScript	Flash	-	GraphML, JSON, Tab delim.	Tutorial, API reference
Gephi	JVM	Java	Swing	MySQL, SQL Server	CSV, DOT, GML, GraphML, GEXL, Tulip, GDF, NET	Wiki, user manual, plugin API reference
Google Vis	Web (JS)	JavaScript	HTML	-	CSV, JSON, XML	User guide, reference
GraphViz	Linux, OSX, Cygwin	C	-	-	DOT	User guides, papers, DOT language spec.
Improvise	JVM	Java	Swing	JDBC	CSV, XML,DBF, SHP	Paper, Ph.D Thesis
IVTK	JVM	Java	Swing	-	CSV, TM3, DOT, TreeML, GraphML	API reference, unfinished user manual
JIT	Web	JavaScript	HTML	-	JSON	Examples/demos w/ source, API reference
JFreeChart	JVM	Java	Swing	JDBC	CSV, XML	Demo app w/ source, API reference, developer guide
JGraph	JVM	Java	Swing	-	GraphML, GD, VDX	Getting started guide, API reference
JUNG	JVM	Java	Swing, SWT	-	GraphML, NET	Tutorial, very short manual, API reference
NetworkX	Python	Python	matplotlib	-	CSV, GML, GraphML, NET	Several tutorials, API reference
Prefuse	JVM	Java	Swing	-	CSV, GraphML, TreeML	API reference, unfinished user manual
Protovis	Web (JS)	JavaScript	HTML	-	-	Tutorials, user guide, API reference
R	Windows, Linux, OSX	C, Fortran, R	-	ODBC	Many	User guides, API reference
Titan	Windows, Linux, OSX	C++, Java, Python, TCL	Qt, TK	MySQL, PostgreSQL, Qt	XGML, VTK, Tulip, HDF5, XML, JSON, CSV, Tab	Wiki, VTK user guide, API references
Tulip	Windows, Linux, OSX	C++	Qt	-	GML, Tulip, DOT	Developer guide, incomplete API reference
VisAD	JVM	Java	Swing	-	Many	Tutorial, incomplete API reference
WilmaScope	JVM	Java	Swing	-	DOT, GML	API reference, plugin developer guide
Zest	JVM	Java	SWT	-	-	Tutorial

REFERENCES

[1] Thomas, J. and Cook, K., [*Illuminating the path: The research and development agenda for visual analytics*], IEEE Computer Society (2005).

[2] Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., and Melançon, G., "Visual analytics: Definition, process, and challenges," in [*Information Visualization*], Kerren, A., Stasko, J., Fekete, J.-D., and North, C., eds., *Lecture Notes in Computer Science* **4950**, 154–175, Springer Berlin / Heidelberg (2008).

[3] Plaisant, C., "The challenge of information visualization evaluation," in [*AVI '04: Proceedings of the working conference on Advanced visual interfaces*], 109–116, ACM, New York, NY, USA (2004).

[4] Shneiderman, B. and Plaisant, C., "Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies," in [*Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*], *BELIV '06*, 1–7, ACM, New York, NY, USA (2006).

[5] Ellis, G. and Dix, A., "An explorative analysis of user evaluation studies in information visualisation," in [*Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*], *BELIV '06*, 1–7, ACM, New York, NY, USA (2006).

[6] Laramee, R. S., "Comparing and evaluating computer graphics and visualization software," *Software: Practice and Experience* **38**(7), 735–760 (2008).

[7] Gonzalez, T. and VanDaniker, M., "Axiis." `http://www.axiis.org/`.

[8] "birdeye information visualization and visual analytics library." `http://code.google.com/p/birdeye/`.

[9] "Flare: Data visualization for the web." `http://flare.prefuse.org/`.

[10] Bastian, M., Heymann, S., and Jacomy, M., "Gephi: An open source software for exploring and manipulating networks," in [*International AAAI Conference on Weblogs and Social Media*], (2009).

[11] "Google Visualization API." `http://code.google.com/apis/charttools/index.html`.

[12] Gansner, E. R. and North, S. C., "An open graph visualization system and its applications to software engineering," *Software: Practice and Experience* **30**(11), 1203–1233 (2000).

[13] Weaver, C. E., *Improvise: a user interface for interactive construction of highly-coordinated visualizations*, PhD thesis, University of Wisconsin - Madison, Madison, WI, USA (2006). Adviser-Livny, Miron.

[14] Fekete, J.-D., "The infovis toolkit," in [*Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*], 167 –174 (2004).

[15] Belmonte, N. G., "JavaScript InfoVis Toolkit." `http://thejit.org/home/`.

[16] "JFreeChart." `http://www.jfree.org/jfreechart/`.

[17] "JGraph." `http://www.jgraph.com/`.

[18] "JUNG - The Java Universal Network/Graph framework." `http://jung.sourceforge.net/`.

[19] Hagberg, A. A., Schult, D. A., and Swart, P. J., "Exploring network structure, dynamics, and function using networkx," in [*Proceedings of the 7th Python in Science Conference*], Varoquaux, G., Vaught, T., and Millman, J., eds., 11–15 (2008).

[20] Heer, J., Card, S. K., and Landay, J. A., "Prefuse: a toolkit for interactive information visualization," in [*CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*], 421–430, ACM, New York, NY, USA (2005).

[21] Bostock, M. and Heer, J., "Protovis: A graphical toolkit for visualization," *Visualization and Computer Graphics, IEEE Transactions on* **15**, 1121–1128 (nov. 2009).

[22] R Development Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria (2010). ISBN 3-900051-07-0.

[23] Wylie, B. and Baumes, J., "A unified toolkit for information and scientific visualizations," in [*Visualization and Data Analysis 2009*], Borner, K. and Park, J., eds., **7243**(1), 72430H, SPIE (2009).

[24] David, A., "Tulip," in [*Graph Drawing*], Mutzel, P., Jünger, M., and Leipert, S., eds., *Lecture Notes in Computer Science* **2265**, 488–491, Springer Berlin / Heidelberg (2002).

[25] Hibbard, B., "Visad: connecting people to computations and people to people," *SIGGRAPH Comput. Graph.* **32**(3), 10–12 (1998).

[26] "Wilmascope 3D graph visualisation system." `http://wilma.sourceforge.net/`.

[27] "Zest: The Eclipse visualization toolkit." `http://www.eclipse.org/gef/zest/`.

[28] Schroeder, W., Martin, K., and Lorensen, B., [*The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*], Kitware, Inc., 4th ed. (December 2006).

[29] Kitware, Inc., [*The VTK User's Guide*], Kitware, Inc., 11th ed. (March 2010).

[30] Shneiderman, B., "The eyes have it: a task by data type taxonomy for information visualizations," in [*Visual Languages, 1996. Proceedings., IEEE Symposium on*], 336–343 (September 1996).

[31] Tory, M. and Möller, T., "Rethinking visualization: A high-level taxonomy," in [*Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*], 151–158 (2004).

[32] Yee, K.-P., Fisher, D., Dhamija, R., and Hearst, M., "Animated exploration of dynamic graphs with radial layout," *Information Visualization, IEEE Symposium on* **0**, 43 (2001).

[33] Buchheim, C., Jünger, M., and Leipert, S., "Improving walker's algorithm to run in linear time," in [*Graph Drawing*], Goodrich, M. and Kobourov, S., eds., *Lecture Notes in Computer Science* **2528**, 347–364, Springer Berlin / Heidelberg (2002).

[34] Fruchterman, T. M. J. and Reingold, E. M., "Graph drawing by force-directed placement," *Software: Practice and Experience* **21**(11), 1129–1164 (1991).

[35] Hu, Y., "Efficient, high-quality force-directed graph drawing," *Mathematica Journal* **10**(1), 37–71 (2005).

[36] Martin, S., Brown, W., Klavans, R., and Boyack, K., "Openord: An open-source toolbox for large graph layout," in [*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*], **7868**, 4 (2011).

[37] Six, J. and Tollis, I., "A framework for circular drawings of networks," in [*Graph Drawing*], Kratochvyl, J., ed., *Lecture Notes in Computer Science* **1731**, 107–116, Springer Berlin / Heidelberg (1999).

[38] Wills, G., "Nicheworks–interactive visualization of very large graphs," in [*Graph Drawing*], 403–414, Springer (1997).

[39] Kamada, T. and Kawai, S., "An algorithm for drawing general undirected graphs," *Information processing letters* **31**(1), 7–15 (1989).

[40] Gansner, E. R., Koutsofios, E., North, S. C., and Vo, K.-P., "A technique for drawing directed graphs," *IEEE Transactions on Software Engineering* **19**(3), 214–230 (1993).

[41] Koren, Y., Carmel, L., and Harel, D., "Ace: A fast multiscale eigenvectors computation for drawing huge graphs," *Information Visualization, IEEE Symposium on* **0**, 137 (2002).

[42] Davidson, R. and Harel, D., "Drawing graphs nicely using simulated annealing," *ACM Transactions on Graphics (TOG)* **15**(4), 301–331 (1996).

[43] Frick, A., Ludwig, A., and Mehldau, H., "A fast adaptive layout algorithm for undirected graphs (extended abstract and system demonstration)," in [*Graph Drawing*], Tamassia, R. and Tollis, I., eds., *Lecture Notes in Computer Science* **894**, 388–403, Springer Berlin / Heidelberg (1995).

[44] Bruls, M., Huizing, K., and van Wijk, J., "Squarified treemaps," in [*In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*], 33–42, IEEE Press (1999).

[45] Melançon, G. and Herman, I., "Circular drawings of rooted trees," *CWI. Information Systems [INS]* (R 9817), 1–13 (1998).

[46] Reingold, E. and Tilford, J., "Tidier drawings of trees," *Software Engineering, IEEE Transactions on* **SE-7**, 223 – 228 (march 1981).

[47] Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., and Henry, N., "Task taxonomy for graph visualization," in [*BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*], 1–5, ACM, New York, NY, USA (2006).

[48] Ousterhout, J., "Scripting: higher level programming for the 21st century," *Computer* **31**, 23–30 (March 1998).

[49] Brandes, U., Eiglsperger, M., Herman, I., Himsolt, M., and Marshall, M., "Graphml progress report structural layer proposal," in [*Graph Drawing*], Mutzel, P., Jünger, M., and Leipert, S., eds., *Lecture Notes in Computer Science* **2265**, 109–112, Springer Berlin / Heidelberg (2002).

[50] Himsolt, M., "GML: A portable Graph File Format," tech. rep., Universität Passau (1997).